

REMARKS/ARGUMENTS

Claims 22-42 were pending and stand rejected in this application. No new matter has been added. Claims 22-42 remain pending in this application. For the reasons set forth below, Applicant respectfully requests that the Examiner reconsider the rejections and allow all of the pending claims.

Obvious-type Double Patenting

Claims 22, 29 and 36 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1 and 9 of co-pending Application No. 10/645,487 to Jameson, U.S. Publication No. US2005/0044095 A1. A terminal disclaimer in compliance with 37 CFR 1.321(c) was filed with the Amendment and Response to Non-Final Action, on January 7, 2008, to overcome the provisional rejection as the conflicting application is shown to be commonly owned with this application. The Examiner notes that the terminal disclaimer is pending. Accordingly the double patenting rejection is upheld until the disclaimer is considered.

Claim Rejections Under 35 U.S.C. 103(a)

On page 4 of the Office Action, claims 22-42 are rejected under 35 U.S.C. §103(a) as being unpatentable over Sundararajan, US 6,487,577 (“Sundaranrajan”).

Independent claims 22, 29 and 36

More specifically, the Office Action asserts that Sundararajan discloses a collection processing operation, “where ‘collection processing operation’ is read on ‘job.’” (Office Action

at page 4, citing Sundararajan at column 7, lines 58-60 and figure 6a, and referring to the term “job”). However, the cited passage does not disclose, teach or suggest this element, only that: “In Fig. 6, the control signal travels from the client computer 604 through the NRS computer 608 to the SC computer 612.”

Additionally, a “job” is limited to compiling, as further described by Sundararajan:

A method of compiling is described. A system for compiling receives a plurality of modules, each module in the plurality of modules representing a portion of a compiling task. The system chooses a plurality of corresponding subcontracting computers to process the plurality of modules based on job-shops generated by the corresponding sub-contracting computers. The system transmits each module as a job to a corresponding sub-contracting computer and combines the output of the sub-contracting computers to generate a compiled task. (*Brief Summary*, Sundararajan at column 2, lines 17-20)

On the other hand, the collection processing operation described by Applicant are not confined to the compilation of modules, but a wide variety of tasks and not so limited as disclosed by the “job” described in Sundararjan. The present application recites:

As part of the dispatching action, the CPS scheduler performs a task name expansion of the first-level symbolic “do this” task name into a sequence of second-level prologue, main, and epilogue processing commands. The CPS scheduler issues these second-level commands to a CPS execution server, one at a time. A typical second-level command sequence, familiar to **software developers who are skilled in the programming arts, might be “checkout my-collection,” “generate makefile,” “make all,” “make install,” and “delete my-collection.** (U.S. Publication No. 20050044067 A1, Paragraph 201 (emphasis added))

Furthermore, as argued in Applicant’s response to Response to Non-Final Action dated January 7, 2008, the collection processing system disclosed by Applicant provides for self-contained information, disclosed as collections, that when delivered to the node computer generate and execute *any* set of commands that are defined by the collection, not limited to compilation. By contrast, a job shop as defined by Sundararajan describes a ‘subcontractor’ computer that accept *only* compilation jobs.

As such, there is a key distinction in the processing of data between the invention of Sudararajan and the invention disclosed by Applicant. The former is limited to the determination of processing order and distribution of compilation of modules. The data transmitted is known to be of one type – compilation, and the task that is sent is to perform a compile. The data sent is the list of modules to compile. In the invention disclosed by Applicant, the Collection Processing System, the data transmitted ‘defines’ the processing to be performed – it contains not just the data to be operated on, as in the case of Sundararajan, but also the definition of the processing to be performed.

Therefore, at the very least, Sundararajan does not disclose, teach or suggest “collection processing operation” as disclosed by Applicant and required in claim 22. Independent claims 29 and 36 also include the element “collection processing operation.”

Dependent claims

Therefore, based at least on the reasons above, Applicant respectfully submits that claims 23-28, which depend from claim 22, claims 30-35 which depend from claim 29, and claims 37-42, which depend from claim 36, are patentable over Sundararajan as applied to claims independent claims 22, 29 and 36.

CONCLUSION

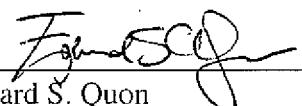
In view of the above, pending claims 22-42 are believed to be in condition for allowance.

On the basis of the above amendments and remarks, reconsideration and allowance of all the pending claims is believed to be warranted and such action is respectfully requested. If it is determined that a telephone conference would expedite the prosecution of this application, the Examiner is invited to telephone the undersigned at the number listed below.

In the event the U.S. Patent and Trademark office determines that an extension and/or other relief is required, Applicant petitions for any required relief including extensions of time and authorizes the Commissioner to charge the cost of such petitions and/or other fees due in connection with the filing of this document to Deposit Account No. 50-1847.

Dated: October 13, 2008

Respectfully submitted,

By 
Edward S. Quon
Registration Number: 52,144
Manatt, Phelps & Phillips, LLP
1001 Page Mill Road - Building 2
Palo Alto, California 94304-1018
Telephone: 650-812-1371
Facsimile: 650-213-0260

20208044.1